PROGRESSIVE TRANSMISSION IN TRELLIS CODED QUANTIZATION-BASED IMAGE CODERS

Philip J. Sementilli[†], *Ali Bilgin, Fang Sheng, Michael W. Marcellin, and John C. Kieffer*[††]

Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ 85721
[†] Science Applications International Corp. (SAIC), 101 N. Wilmot Rd., Tucson, AZ 85711
[††]Department of Electrical Engineering, University of Minnesota, Minneapolis, MN 55455

ABSTRACT

As image coders evolve from DCT-based to wavelet-based designs, the latter must be enhanced to include capabilities currently supported by standards such as JPEG. Recent work [1, 2, 3] has described approaches for incorporating progressive transmission capabilities within wavelet-based coders. All of these coders apply scalar quantization to wavelet transform coefficients, and then apply sophisticated entropy coding methods to the quantized coefficients. In this paper, we present coding techniques that enable progressive transmission when trellis coded quantization (TCQ) is applied to the wavelet coefficients. While the trellis coded quantizer is more complex than the uniform scalar quantizer, comparable PSNR performance is achieved with a simple entropy coder. In addition, our use of sophisticated quantization and bit rate allocation algorithms enables the development of coders that are tuned for improved perceptual image quality.

1. INTRODUCTION

JPEG, the current ISO standard for still image compression [4], includes two modes for incremental decoding of imagery, hierarchical and progressive. The hierarchical mode yields improved resolution as more bits are decoded. The progressively decodable modes successively increase pixel accuracy in the decoded image. JPEG defines spectral selection (successive decoding from low- to high-frequency DCT coefficients) and successive approximation (successive decoding from most significant to least significant bit of the DCT coefficients) progressive modes. These modes add desireable features to the image coder, such as the capability for image browsing and progressive transmission, and natural prioritization for layered protection schemes.

It has long been realized that wavelet-based methods are naturally suited to hierarchical transmission. Recent works ([1, 2, 3]) have described approaches for incorporating progressive transmission capabilities within wavelet-based coders. These coders apply scalar quantization to the wavelet coefficients, and then apply sophisticated entropy coding methods to the quantized coefficients.

This paper describes successive approximation-type methods for use with a wavelet/TCQ (WTCQ) coder. We investigate the performance of the WTCQ coder with different bit rate allocation and entropy coding techniques. Comparisons with other results from the literature show that our system is quite competitive.

2. CODER DESCRIPTION

A simplified block diagram of the WTCQ coder is shown in Figure 1. The wavelet coefficients in each subband are scanned into "sequences." The rate allocator determines the step size to be used for each sequence, and the sequences are quantized using TCQ. Finally, the TCQ indices are entropy coded. Notice that the block diagram of Figure 1 does not specify what type of rate allocator and entropy coder are used, since we use different techniques for these blocks and compare the results.

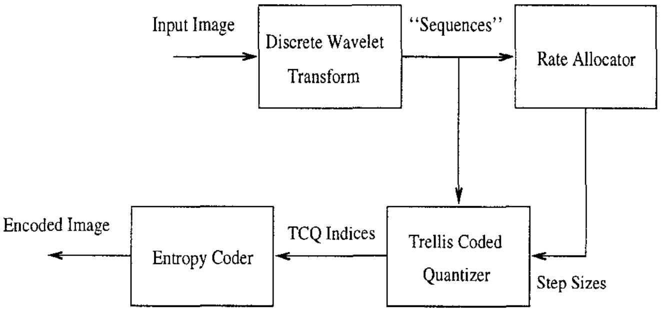

Figure 1: Block Diagram of the WTCQ coder.

THIS WORK WAS SUPPORTED IN PART BY THE NATIONAL SCIENCE FOUNDATION UNDER GRANT NO. NCR-9258374

0-8186-8183-7/97 $10.00 © 1997 IEEE

We have considered two different approaches for the rate allocator. In the first case, the rate allocator determines a single step size to be used in every sequence, such that the total rate does not exceed the target rate. In the second case, the step size is allowed to vary on a sequence by sequence basis, and the rate allocation procedure described in [5] is used to determine the optimum step sizes.

The output from the trellis coded quantizer is a set of signed integers describing a path through the trellis. Previous versions of the WTCQ coder [6] used M-ary arithmetic coding of these indices, yielding PSNR performance comparable to the best coders developed at that time. In this paper, we describe alternatives to the M-ary arithmetic coder that enable progressive transmission. To implement a successive approximation-type embedded code, the M-ary arithmetic coder is replaced with either of two entropy coders; a binary arithmetic coder that encodes the set of TCQ indices from most significant to least significant bit (i.e., a bit plane coder), or the hierarchical tree-based method used in the SPIHT coder [1].

2.1. WTCQ with Bit Plane Coder

For an embedded successive approximation code, the bit planes are ordered so that all bits at a given power of two for all sequences are transmitted, followed by all bits at the next lower power of two, etc. The adaptive binary arithmetic coder described in [7] is used to encode the bit sequences. The coder is made more efficient by the use of multiple contexts when accumulating frequency histograms. These contexts are derived from neighboring TCQ indices within a sequence. The sign bit for each TCQ index is encoded in the bit stream immediately after encoding the most significant bit of each index. The arithmetic coder frequency histograms are re-initialized each octave, to capture dependencies across subbands at each level of the wavelet tree.

2.2. WTCQ with SPIHT Entropy Coder

As presented in [1], SPIHT is a scalar quantizer based technique. However TCQ can be used to replace the implicit scalar quantization. Moving from the lowest subband to the highest, the wavelet coefficients in each subband are raster scanned and trellis coded quantized. Either of the two rate allocators can be used to determine the TCQ step sizes. The resulting signed integer image is entropy coded using SPIHT. This approach was also used in [8]. It is important to note that the image can be recovered progressively up to a fully decoded file corresponding to the chosen step size(s).

3. PROGRESSIVE TCQ

Figure 2 shows a 4-state trellis. In TCQ, for a given sequence of data, the Viterbi algorithm is used to pick the allowed trellis path that minimizes the mean-squared error between the input data and the output codewords. In any given trellis state, the output codeword is selected from one of two *supersets* $S_0 = D_0 \bigcup D_2$ or $S_1 = D_1 \bigcup D_3$ (See Figure 3).

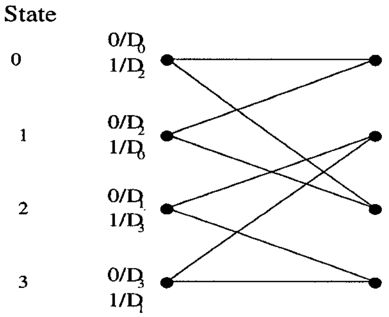

Figure 2: Four State Trellis.

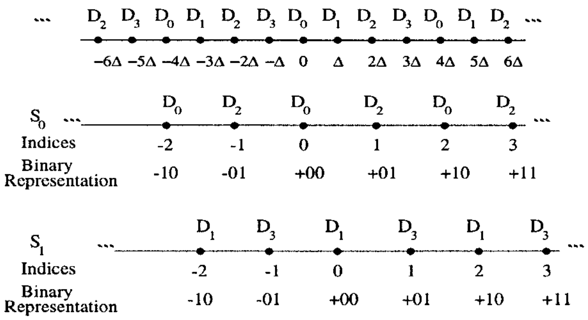

Figure 3: TCQ Indices.

The codeword index of the superset is enough to determine the subset the codeword was selected from, since the supersets are disjoint. Thus the next trellis state is also determined by this index and the output codewords are represented by a sequence of signed integers.

It can be seen in Figure 3 that the codewords of the two subsets of a superset differ in the least significant bit. In other words, the least significant bit of the codeword index determines which subset the codeword belongs to. The least significant bit thus determines the path through the trellis. Since the decoder needs to be able to determine the path, it is not possible to perfectly invert the TCQ indices until all of the least significant bits are available. For this reason, in a successive approximation-type setting, we perform the inverse TCQ operation only approximately until all of an encoded sequence has been received. The signed integer consisting of the bits that have been received so far is multiplied by twice the step size to approximate the inverse TCQ operation. When the bits for an entire sequence are received, the full inverse TCQ operation is performed on that sequence.

4. CODER PERFORMANCE

We have compared the performance of the presented system using different rate allocation schemes and entropy coders. We constructed four different architectures, namely, OTA (Optimum step sizes / TCQ / Arithmetic bitplane coder), CTA (Constant step sizes / TCQ / Arithmetic bitplane coder), OTS (Optimum step sizes / TCQ / SPIHT coder), and CTS (Constant step sizes / TCQ / SPIHT coder). Figure 4 compares the PSNR performance of these systems on Goldhill. All coders use the (9,7) biorthogonal filter bank of [9] and have 3 levels of uniform decomposition, followed by 2 more levels of dyadic decomposition of the reference band ("3+2"). Our tests show that OTA performs better than the other algorithms.

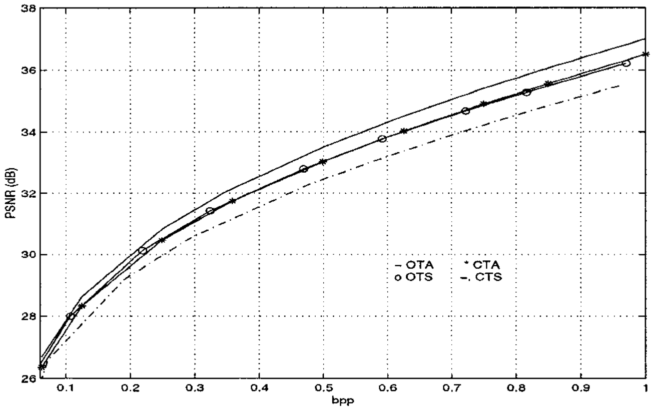

Figure 4: Performance of presented algorithms on Goldhill.

Tables 1 and 2 show PSNR performance of OTA and SPIHT [1] on Barbara and Goldhill, respectively. These results are for full decoding of the progressively decodable files. All results use the (9,7) filters, with either 3 levels of uniform decomposition followed by m levels of dyadic decomposition of the reference band ("3+m"), or a k level dyadic decomposition.

Figure 5 illustrates the progressive performance of OTA on the Barbara image. The solid line shows the performance of fully decoded OTA. This is compared to the progressive decoding of two OTA files, one coded at a rate of 0.5 bpp and the other coded at 1.0 bpp. Note that the image quality (PSNR) is acceptable at low decoding rates, and that PSNR monotonically improves as more bits are decoded. The progressive performance of the coder starts to degrade towards the end of the file. But once the path through the trellis is determined, i.e. the least significant bits of the quantization indices are received, the performance of the coder improves dramatically.

Table 1: PSNR results for fully decoded Barbara image using OTA and SPIHT coders

Method	Decomposition	PSNR (dB)		
		0.25 bpp	0.5 bpp	1.0 bpp
OTA	3+2	29.16	32.90	37.63
SPIHT	3+3	29.00	32.73	-
OTA	5-level dyadic	27.85	31.50	36.57
SPIHT	6-level dyadic	27.58	31.39	36.41

Table 2: PSNR results for fully decoded Goldhill image using OTA and SPIHT coders

Method	Decomposition	PSNR (dB)		
		0.25 bpp	0.5 bpp	1.0 bpp
OTA	3+2	30.83	33.51	37.01
SPIHT	3+3	30.69	33.17	-
OTA	5-level dyadic	30.65	33.34	36.93
SPIHT	6-level dyadic	30.56	33.13	36.55

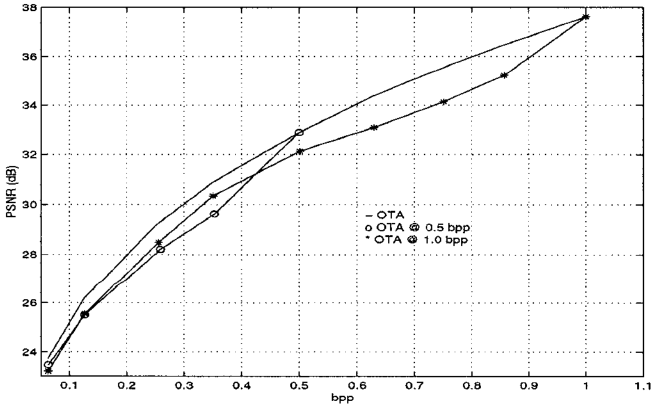

Figure 5: Progressive Performances of OTA on Barbara.

---

[1] The results for SPIHT using (9,7) filters and 3+3 decomposition were obtained from *http://www.icsl.ucla.edu/~ipl/index.html*

5. SUMMARY

We have shown that wavelet-TCQ image coders can be configured to support hierarchical and progressive modes analogous to those specified by the JPEG standard. These modes can be supported with little or no loss in coding efficiency. We have shown that a relatively simple entropy coder combined with a sophisticated quantizer results in a image coder with PSNR performance similar to that achieved using a simple quantizer and sophisticated entropy coder.

6. REFERENCES

[1] A. Said and W.A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees", *IEEE Transactions on Circuits and Systems for Video Tech.*, vol. 6, pp. 243–250, June 1996.

[2] E. L. Schwartz, A. Zandi, and M. Boliek, "Implementation of compression with reversible embedded wavelets", in *Proc. of SPIE*, July 1995, vol. 2564.

[3] J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients", *IEEE Transactions on Signal Processing*, vol. 41, pp. 3445–3462, Dec. 1993.

[4] W. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Data Compression Standard*, Van Nostrand Reinhold, New York, 1992.

[5] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers", *IEEE Transactions on Acous., Speech, and Signal Processing*, vol. ASSP-36, pp. 1445–1453, Sept. 1988.

[6] J. H. Kasner and M. W. Marcellin, "Adaptive wavelet coding of images", in *Proc. of 1994 Int. Conf. on Image Processing*, Nov. 1994, vol. 3, pp. 358–362.

[7] I. H. Witten, R. M. Neal, and J.G. Cleary, "Arithmetic coding for data compression", *Communications of the ACM*, vol. 30, pp. 520–540, June 1987.

[8] W. Zhang and T. R. Fischer, "TCQ subband image coding to exploit dependence", in *Proc. of 1996 Int. Conf. on Image Processing*, Sept. 1996, vol. 2, pp. 553–556.

[9] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform", *IEEE Transactions on Image Processing*, vol. 1, pp. 205–220, Apr. 1992.